EXPRESS MAILING LABEL
NO. EV371747165US

174/329
(A1393)

MAINTAINING DATA INTEGRITY FOR EXTENDED DROP OUTS ACROSS HIGH-SPEED SERIAL LINKS

Background of the Invention

[0001] This invention relates to error correction circuitry and more particularly to error correction circuitry that can provide data recovery during extended drop out periods of a high speed serial link with an embedded clock signal.

[0002] Typical error correction techniques, such as, for example, forward error correction (FEC), allow a receiving device to detect and correct corrupted data that contains fewer than a predetermined number of errors. FEC techniques add extra bits or bytes to each transmitted data character or code block using a predetermined algorithm.

[0003] An increasingly important type of signaling between devices is serial signaling in which the clock signal information is embedded in or can be derived from the serial data stream so that no separate clock signal needs to be transmitted. For example, data may be transmitted serially in packets of several successive serial data words preceded by a serial

header that includes several training bits having a predetermined pattern of binary ones and zeros. The clock signal information is embedded in the data signal by the high-to-low and/or low-to-high transitions in that signal, which must have at least one high-to-low or low-to-high transition within a certain number of clock signal cycles. At the receiver, the clock signal is recovered from the data signal for use in properly processing the data signal. For convenience herein, this general type of signaling will be referred to generically as "clock data recovery" or "CDR" signaling.

[0004] CDR signaling is now being used in many different signaling protocols. These protocols vary with respect to such parameters as clock signal frequency, header configuration, packet size, data word length, number of parallel channels, etc. CDR signaling is well known as shown, for example, in Aung et al. U.S. Patent Publication No. 2001/0033188, published October 25, 2001 and Aung et al. U.S. Patent Publication No. 2003/0212930, published November 13, 2003.

[0005] Ordinarily, FEC transmission techniques are able to provide data correction for extended data errors across data links. However, when a reference clock signal is embedded within a serial data stream, extended data errors may also result in a loss of the reference clock signal and a loss of synchronization between the transmitter and the receiver. This loss of synchronization limits the effectiveness of typical FEC techniques.

[0006] Programmable logic devices ("PLDs") are well known as shown, for example, by such references as

Cliff et al. U.S. Patent 5,689,195, Cliff et al. U.S. Patent 5,909,126, Jefferson et al. U.S. Patent 6,215,326, and Ngai et al. U.S. Patent 6,407,576. In general, a PLD is a general-purpose integrated circuit device that is programmable to perform any of a wide range of logic tasks. Rather than having to design and build separate logic circuits for performing different logic tasks, general-purpose PLDs can be programmed in various different ways to perform those various logic tasks. Many manufacturers of electronic circuitry and systems find PLDs to be an advantageous way to provide various components of what they need to produce.

[0007] It would be highly desirable to have the ability to use PLDs to avoid having to always design and build error correcting transmitters and receivers that are specific to each of the many different error correction techniques, protocols, and specifications.

Summary of the Invention

[0008] In accordance with this invention, improved error correction techniques and circuitry are provided. The circuitry of this invention is preferably programmable in at least some respects and may either be included in the I/O circuitry of an integrated circuit with other more traditional PLD circuitry, or with other non-programmable circuitry. It may also be at least partly included on a separate integrated circuit. If the data correction circuitry is at least partly on a separate circuit, it may be configured to facilitate efficient coupling to other circuitry (e.g., in a common package).

[0009] Error correction techniques and circuitry in accordance with the invention preferably use

traditional FEC data frames and break the FEC frames into sub-packets. Before each sub-packet is transmitted, an IDLE/SYNC packet is sent. Thus each received FEC frame will contain multiple data sub-packets, each proceeded by an IDLE/SYNC packet.

[0010] Normally, after an extended loss of the transmitted data (e.g., a data drop off), the data receiver loses synchronization with the data transmitter. Thus, after data loss, the position of the incoming valid data would be unknown relative to the last valid data received prior to the data loss. With gaps or holes in the data of undeterminable size, FEC techniques would be unable to recover the lost data. However, according to the present invention, the IDLE/SYNC packets may be used by the data receiver to determine the relative position of the received data and allow the traditional error correction techniques to function properly. After the data receiver detects a data drop off or another type of error in the transmitted data, the receiver will ignore the remaining incoming data until it receives an IDLE/SYNC packet. The information contained in the IDLE/SYNC packet allows the data receiver to determine the relative position of the following data sub-packet. The FEC system may then be able to correct the errors in the FEC data frame.

[0011] Various characteristics of this data (e.g., frame size, packet size, word length, etc.) are preferably selectable (e.g., programmable). In addition, the error correcting receiver and/or transmitter circuitry of this invention may also be programmable in other respects. For example, the error correcting circuitry may implement various error

correcting algorithms with programmable operating parameters which may be adjusted or optimized based on the nature of the data transmitted, the transmission link, or other system parameters. The error correcting circuitry may also include the capability of operating selectable numbers of error correcting data receiver and/or transmitter subcircuits in parallel.

[0012] The circuitry of this invention may also be programmable to alternatively support other types of non-CDR signaling such as non-CDR low-voltage differential signaling ("LVDS"). The circuitry of this invention may be constructed to provide signals such as loss-of-lock and run-length violation signals that can be used as indications that various parts of the circuitry need to be reset. Circuitry for facilitating reset and/or power down of various portions of the circuitry can also be provided. Circuitry for selectively creating various types of test loops in the circuitry may be provided to facilitate testing various portions of the circuitry. Circuitry for programmably modifying a reference clock signal in certain modes of operation (especially a reference clock signal output by the programmable logic device) may also be provided.

[0013] Because the invention facilitates handling data with a PLD, the logic of the PLD can be used to manipulate the data in accordance with whatever protocol is being used (e.g., with respect to such aspects as byte alignment, comma detect, word length, or any other aspect of decoding the data on the receiver side and/or encoding the data on the transmitter side). The present combination of error correction techniques and PLD circuitry is therefore very advantageous.

Brief Description of the Drawings

[0014] The above and other objects and advantages of the invention will be apparent upon consideration of the following detailed description, taken in conjunction with the accompanying drawings, in which like reference characters refer to like parts throughout, and in which:

[0015] FIG. 1 is a simplified block diagram of an illustrative embodiment of a typical forward error correction (FEC) system;

[0016] FIG. 2 is a simplified block diagram of an illustrative embodiment of a typical FEC system having an embedded clock signal;

[0017] FIG. 3 is a simplified block diagram of an illustrative embodiment of an improved FEC system having an embedded clock signal in accordance with the invention;

[0018] FIG. 4 is a simplified diagram of an illustrative data packet in accordance with the invention;

[0019] FIG. 5 is a simplified diagram of an illustrative FEC data frame in accordance with the invention;

[0020] FIG. 6 is a simplified schematic block diagram of an illustrative embodiment of the transmission circuitry of an improved FEC system having an embedded clock signal in accordance with the invention;

[0021] FIG. 7 is a simplified schematic block diagram of an illustrative embodiment of the receiver circuitry of an improved FEC system having an embedded clock signal in accordance with the invention; and

[0022] FIG. 8 is a simplified block diagram of an illustrative system employing circuitry in accordance with the invention.

Detailed Description of the Invention

[0023] FIG. 1 shows an illustrative diagram of existing forward error correction (FEC) system 100. FEC system 100 may allow data corrupted during transmission to be corrected without requiring retransmission of the data. In FEC system 100, data is encoded by FEC encoder 110 and then transmitted over transmission channel 150 to FEC decoder 160 which decodes and outputs the received data.

[0024] Exemplary FEC encoder 110 may contain Reed-Solomon encoder 120 and symbol interleaver 130. Reed-Solomon code is a data correction algorithm commonly used in FEC techniques. A Reed-Solomon encoder converts a stream of digital data into a number of information codewords adding additional "redundant" bits. A Reed-Solomon decoder may then attempt to correct errors in the received data in order to recover the original data stream. The number and type of errors that can be corrected depends on the characteristics and parameters of the Reed-Solomon code, such as the size of the codewords and the number of additional bits. The Reed-Solomon algorithm and the circuitry used to implement Reed-Solomon encoders and decoders are well known and are commonly used to provide error detection and correction.

[0025] The output of Reed-Solomon encoder 120 may then be input to symbol interleaver 130. Symbol interleavers and de-interleavers are also well known and may be used to mitigate the effects of multiple,

successive data errors in a data stream. These types of data errors can be caused by occurrences such as an extended data drop-off or sudden periods of error inducing noise (e.g., burst noise). An interleaver takes a data stream and modifies or "mixes-up" the order of the bits, bytes, words, or packets within the data stream according to a known algorithm. Interleaving a Reed-Solomon-encoded data stream improves the efficiency of Reed-Solomon encoders/decoders by spreading burst errors across several or many Reed-Solomon codewords instead of being localized in one or a few codes. The output of symbol interleaver 130 is an FEC frame having a set number of bits, including the data and "redundant" bits if the Reed-Solomon codeword length does not divide evenly into the interleaver frame size. The size of the FEC frame and the specific algorithm used or the parameters used to implement the algorithm, may preferably be adjustable or programmable or may be determined by the parameters of Reed-Solomon encoder 120 or symbol interleaver 130, which also may be adjustable or programmable. The encoded and interleaved data can then be transmitted and de-interleaved and decoded at the data receiver with FEC decoder 160 having symbol de-interleaver 170 and Reed-Solomon decoder 180.

[0026] FEC system 100 is effective for providing data correction and recovery in the presence of brief data drop-outs across a transmission channel. However, FEC system 100 is only effective for providing data correction and recovery for extended data drop-outs in situations where there is a reference clock separate from the transmitted data stream. With the separate reference clock, the data recovery circuitry may easily

detect the relative data position and the length of the data drop-out in order to facilitate the error correction and recovery. Although the typical FEC system 100 described herein uses Reed-Solomon coding techniques in combination with interleaving/de-interleaving techniques, other known FEC techniques may be used in accordance with the present invention.

[0027] FIG. 2 shows an illustration of typical FEC system 200 having an embedded clock signal. FEC system 200 contains FEC encoder 210, transmission channel 250, and FEC decoder 260 which all operate similarly or identically to their corresponding counterparts in FEC system 100. FEC system 200 also preferably include data transmitter 240 and data receiver 290. Data transmitter 240 and data receiver 290 preferably includes clock data recovery (CDR) transmitter and receiver circuitry, respectively, which allow a reference clock to be embedded in, and recovered from, a data stream. The embedded reference clock of the CDR data signal can be the same frequency as the reference clock frequency or any convenient fraction or multiple of the embedded clock frequency. In particular, the reference clock signal frequency REFCLK is related to the embedded clock frequency EMBCLK by the following relationship:

$$REFCLK*W = EMBCLK,$$

where W is a convenient scale factor such as 0.5, 1, 2, 4, etc.

[0028] FEC system 200 may only be able to recover the data loss during limited data drop-outs. However, after an extended data drop out, the reference clock embedded within the data signal will be lost and thus

the synchronization between data transmitter 240 and data receiver 290 will also be lost. Under these conditions the traditional FEC technique of FEC system 200 will not provide adequate data correction.

[0029] FIG. 3 illustrates an improved FEC error correction system 300 which may be able to maintain data integrity for extended drop outs across high speed serial links. FEC system 300 includes FEC encoder 310, data transmitter 340, transmission channel 350, data receiver 390, and FEC decoder 360 which all operate similarly or identically to their corresponding counterparts in FEC system 200. FEC system 300 also preferably includes channel SYNC encoder 345 and channel SYNC decoder 355.

[0030] Channel SYNC encoder 345 breaks each FEC data frame into sub-packets and inserts an IDLE/SYNC packet before each FEC frame sub-packet. FIG. 4 shows illustrative FEC data packet 400 having data sub-packet 410 which may be, for example, 16 double words in length along with an IDLE/SYNC packet 420 which may be, for example, one double word in length. In this example, each IDLE/SYNC packet is made up of two 8 bit data words. The first word of the IDLE/SYNC character is used to provide the IDLE character and the other word is used for the SYNC character. The length of the packets of the FEC frame sub-packets and the IDLE/SYNC packets used in the present example are merely illustrative. The size of these packets and the contents and make-up of the IDLE/SYNC packets may be adjustable or programmable. FIG. 5 shows illustrative FEC frame 500 output from channel SYNC encoder 345 according to the previous example. Frame 500 is made

up of 100 data sub-packets 410 and 100 IDLE/SYNC packets 420.

[0031] The IDLE character of IDLE/SYNC packets may be used to keep data receiver 360 aligned in the boundaries of the FEC data frame sub-packets 410. The following example illustrates one technique by which the IDLE/SYNC packet may be used to maintain data alignment. Channel SYNC decoder 355 may contain two control signals, IDLE detect and IDLE status (not shown). The IDLE detect signal indicates when each IDLE signal has been detected by channel SYNC decoder 355. The IDLE status signal indicates when an IDLE character has been found by channel SYNC decoder 355 in an unexpected or incorrect location in the received data stream. Using these signals and a counter (also not shown), it is possible to determine that data has lost synchronization with the clock signal or that data have been lost or misaligned. Assuming the system is sending error free packets as shown in FIG. 4, the IDLE detect signal will be asserted every 17 packets received. If the IDLE detect signal is not asserted, one or more bits or bytes has been lost or a transmission error has occurred. Similarly, if the IDLE status signal is asserted, this also may indicate that one or more bits or bytes has been lost or corrupted. If either of these things happen, the received data is not written to memory. The system will wait until the next valid IDLE character is received.

[0032] The SYNC character of the IDLE/SYNC packet is used to embed control information within the FEC data frame. The SYNC character contains a number or a character, such as a sub-packet identification (ID)

number, that may indicate the proper position of the following data sub-packet within the FEC frame. This sub-packet ID number can be used to determine the relative or absolute position of each data sub-packet within the FEC data frame.

[0033] For example, in the event of a transmission error, the position of an FEC frame sub-packet received following a valid IDLE/SYNC packet may be determined based on the packet ID number contained within the IDLE/SYNC packet. The FEC technique may allow the sub-packets lost or corrupted due to transmission errors to be recovered by FEC decoder 360.

[0034] FIG. 6 shows an illustrative schematic of the transmission circuitry of an improved FEC error correction system in accordance with the invention. FEC error correction system 600 includes FEC encoder 610, data transmitter 640, and channel SYNC encoder 645 which all operate similarly or identically to their corresponding counterparts in FEC system 300. FEC encoding circuitry 610 includes Reed-Solomon encoder 620 and symbol interleaver 630, which both operate similarly or identically to their corresponding counterparts in FEC system 100.

[0035] In the present embodiment FEC system 600 receives an input which includes 16 parallel data inputs. This number of parallel data inputs is merely illustrative of a single embodiment of the present invention. According to another embodiment of the invention, any number of parallel data inputs or even a single data input may also be used. The data input to FEC system 600 is input to FEC encoding circuitry 610. At the input of FEC encoding circuitry 610 is first-in/first-out (FIFO) memory buffer 611. FIFO 611

receives the data input signals and outputs the data to Reed-Solomon encoder 620. FIFO 611 is used to properly control the flow of the data from the input of FEC encoder 610 to the input of Reed-Solomon encoder 620. FIFO 611 is used because the data input to FEC encoding circuitry 610 may be received at a different data rate than may be sent to the input of Reed-Solomon encoder 620. The data rates of the various elements of FEC system 600 are preferably adjustable or programmable or may be based on the preferably adjustable or programmable system specifications of FEC system 600.

[0036] The outputs of FIFO 611 are input to Reed-Solomon encoding circuitry 622, which is represented in the present embodiment by Reed-Solomon encoding circuitry 622a and Reed-Solomon encoding circuitry 622b. In this embodiment, two Reed-Solomon encoding circuits, each having 8 parallel inputs, are used to encode the 16 parallel data inputs. Any suitable number of conventional Reed-Solomon encoding circuits, having any suitable number of data inputs, may be used based on the design of the particular FEC encoding system. Each of the Reed-Solomon encoding circuits are controlled by Reed-Solomon controller 624 and output Reed-Solomon data packets. The outputs of Reed-Solomon encoder 620 are connected to the inputs of symbol interleaver 630.

[0037] Symbol interleaver 630 includes FIFO 631 to receive the outputs of Reed-Solomon encoder 620. Like FIFO 611, FIFO 631 can be used to support differences in the data rates between the output rate of Reed-Solomon encoder 620 and the input rate of symbol interleaver 630. Symbol interleaver 630 also includes

multiplexer 632, interleaver 634, and interleaver framing control 636. Multiplexer 632 may be switched to alternatively pass the Reed-Solomon data packets interleaver framing control 636 packets to the input of interleaver 634. Interleaver framing control 636 may switch multiplexer 632 in order to insert additional header data and any necessary padding bits to form the FEC data frame at the outputs of symbol interleaver 634. The outputs of symbol interleaver 634 are connected to the inputs of channel SYNC encoder 645.

[0038] Channel SYNC encoder 645 includes FIFO 646 to receive the outputs of symbol interleaver 630. Like FIFO 611 and FIFO 631, FIFO 646 can be used to support different input and output data rates. Channel SYNC encoder 645 also includes multiplexer 647 and channel SYNC controller 648. As the FEC data packets are output from FIFO 646, channel SYNC controller 648 can switch multiplexer 647 to insert IDLE/SYNC packets into the FEC data packets at the appropriate intervals. The outputs of channel SYNC encoder 645 are connected to the inputs of data transmitter 640.

[0039] Data transmitter 640 may be any suitable or conventional data transmitter. According the present embodiment of the invention, data transmitter additionally includes 8B10B encoder 642 and serializer 644. The outputs of channel SYNC encoder 645 are connected to data transmitter 640. Data transmitter 640 may use 8B10B encoder 642 to encode every 8-bits of a data stream into 10-bit codes that may result in a DC-balanced bit stream (e.g., the number of 0's and the number of 1's are equal), ease receiver synchronization, and may be able to insert and

detect special control characters. Data transmitter 640 may also include any other type of suitable data encoder. Data transmitter 640 may also have serializer 644 which may convert the parallel inputs received by the transmitter into a single serial output.

[0040] FIG. 7 shows an illustrative schematic of the receiver circuitry 700 of an improved FEC error correction system in accordance with the invention. FEC receiver circuitry 700 includes data receiver 790, channel SYNC decoder 755, and FEC decoder 760 which all operate similarly or identically to their corresponding counterparts in FEC system 300. FEC decoder 760 includes Reed-Solomon decoder 780 and symbol de-interleaver 770, which both operate similarly or identically to their corresponding counterparts in illustrative FEC system 100.

[0041] Data receiver 790 may be any suitable or conventional data receiver which is configured to receive the transmitted data signal from the data transmitter 640. According to the present embodiment data receiver 790 additionally includes 8B10B decoder 792 and de-serializer 794. The serial signal received at data receiver 790 is decoded and converted to multiple parallel signals, if necessary, and is output to channel SYNC decoder 755. Channel SYNC decoder 755 includes channel SYNC control 756 and frame storage memory 758. As the data signal or signals are output from the data receiver, channel SYNC control 756 detects the IDLE/SYNC packets. As previously described, during error-free transmission each FEC frame sub-packet is preceded by an IDLE/SYNC packet. When channel SYNC control 756 detects the IDLE/SYNC

packets, channel SYNC control 756 writes the data of the following FEC frame sub-packet to the proper location of frame storage memory 758. Sub-packet identification (ID) number of the IDLE/SYNC packet can be used by channel SYNC control 756 to direct the received data to the proper memory location in frame storage memory 758. If an IDLE/SYNC packet is not received in the proper location in the data stream, the incoming data is not written to frame storage memory 758 until after the next valid IDLE/SYNC packet is received.

[0042] After the received data is written to frame storage memory 758, the data is decoded by FEC decoder 760 which includes Reed-Solomon decoder 780 and symbol de-interleaver 770, which operate in a similar manner as their encoding counterparts in FIG. 6.

[0043] FIG. 8 illustrates a PLD or multi-chip module 800 of this invention in a data processing system 1002. Data processing system 1002 may include one or more of the following components: a processor 1004; memory 1006; I/O circuitry 1008; and peripheral devices 1010. These components are coupled together by a system bus or other interconnections 1020 and are populated on a circuit board 1030 which is contained in an end-user system 1040. Any of the interconnections between element 800 and any other elements may be made using the above-described improved FEC signaling techniques.

[0044] System 1002 can be used in a wide variety of applications, such as computer networking, data networking, instrumentation, video processing, digital signal processing, or any other application where the advantage of using programmable or reprogrammable logic is desirable. PLD/module 800 can be used to perform a

variety of different logic functions. For example, PLD/module 800 can be configured as a processor or controller that works in cooperation with processor 1004. PLD/module 800 may also be used as an arbiter for arbitrating access to a shared resource in system 1002. In yet another example, PLD/module 800 can be configured as an interface between processor 1004 and one of the other components in system 1002. It should be noted that system 1002 is only exemplary, and that the true scope and spirit of the invention should be indicated by the following claims.

[0045] Various technologies can be used to implement PLDs or multi-chip modules 800 having the features of this invention, as well as the various components of those devices (e.g., the above-described PLCs and programmable function control elements ("FCEs") that control the PLCs). For example, each PLC can be a relatively simple programmable connector such as a switch or a plurality of switches for connecting any one of several inputs to an output. Alternatively, each PLC can be a somewhat more complex element that is capable of performing logic (e.g., by logically combining several of its inputs) as well as making a connection. In the latter case, for example, each PLC can be product term logic, implementing functions such as AND, NAND, OR, or NOR. Examples of components suitable for implementing PLCs are EPROMs, EEPROMs, pass transistors, transmission gates, antifuses, laser fuses, metal optional links, etc. PLCs and other circuit components can be controlled by various, programmable, function control elements ("FCEs"). (With certain implementations (e.g., fuses and metal optional links) separate FCE devices are not required.)

FCEs can also be implemented in any of several different ways. For example, FCEs can be SRAMs, DRAMs, first-in first-out ("FIFO") memories, EPROMs, EEPROMs, function control registers (e.g., as in Wahlstrom U.S. Patent 3,473,160), ferro-electric memories, fuses, antifuses, or the like. From the various examples mentioned above it will be seen that this invention is applicable to both one-time-only programmable and reprogrammable devices.

[0046] It will be understood that the foregoing is only illustrative of the principles of this invention, and that various modifications can be made by those skilled in the art without departing from the scope and spirit of the invention. For example, the numbers of the various types of resources and components can be different from the numbers present in the depicted and described illustrative embodiments.